(19)日本国特許庁（ＪＰ）　(12)公表特許公報（Ａ）　(11)特許出願公表番号
特表2002－533787
（P2002－533787A）
(43)公表日　平成14年10月８日(2002.10.8)

| (51)Int.Cl.⁷ | 識別記号 | ＦＩ | テーマコード（参考） |
|---|---|---|---|
| Ｇ０９Ｃ　1/00 | ６２０ | Ｇ０９Ｃ　1/00 | ６２０Ａ　５Ｊ１０４<br>６２０Ｚ |

審査請求　未請求　予備審査請求　有　（全 50 頁）

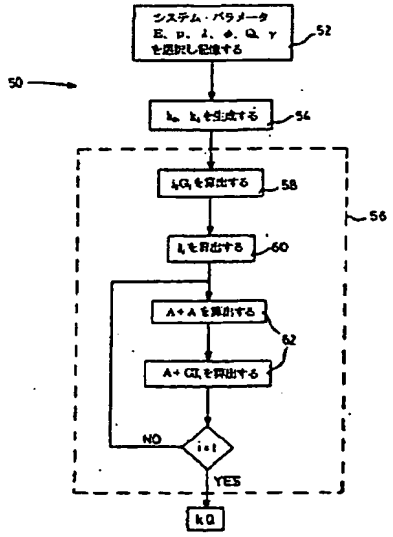
(21)出願番号　特願2000－591498(P2000－591498)
(86)(22)出願日　平成11年12月23日(1999.12.23)
(85)翻訳文提出日　平成13年６月22日(2001.6.22)
(86)国際出願番号　ＰＣＴ／ＣＡ９９／０１２２２
(87)国際公開番号　ＷＯ００／３９６６８
(87)国際公開日　平成12年７月６日(2000.7.6)
(31)優先権主張番号　２，２５７，００８
(32)優先日　平成10年12月24日(1998.12.24)
(33)優先権主張国　カナダ（ＣＡ）

(71)出願人　サーティコム　コーポレーション
カナダ，オンタリオ州　エル４ダブリュー　５エル１，ミシソーガ　フォース　フロア，エクスプローラー　ドライブ　5520
(72)発明者　ガラント、ロバート
カナダ国　エル５エム　５エヌ１　オンタリオ州、ミシソーガ、ローズブッシュ　ロード　4788
(72)発明者　ランバート、ロバート、ジェイ.
カナダ国　エヌ３シー　３エヌ３　オンタリオ州、ケンブリッジ、ホルム　ストリート　63
(74)代理人　弁理士　平木　祐輔　（外２名）

最終頁に続く

(54)【発明の名称】　楕円曲線上での暗号操作の速度を高める方法

(57)【要約】
本発明は、楕円曲線点Q(x，y)とスカラーｋとの乗算の速度を高める方法であって、ｑが、楕円曲線上のすべての点Q(x，y)についてψ(Q) = λ・Qである自己準同型写像ψが存在するような素数べき乗である、有限体Fqの上の楕円曲線を選択するステップと、スカラーｋのより小さな表現$k_i$と写像ψの組合せを使用して楕円曲線点Qのスカラー倍数を算出するステップとを含む方法を提供する。



【特許請求の範囲】

【請求項１】　楕円曲線点Q(x, y)にスカラーを乗じて点kQを提供する方法であって、

ａ）λを整数として、楕円曲線上のすべての点Q(x, y)についてψ(Q) = λ・Qである自己準同形写像ψが存在するように、有限体Ｆの上の楕円曲線を選択するステップと、

ｂ）前記スカラーｋの表現を成分$k_i$と前記整数λの組合わせとして確立するステップと、

ｃ）前記表現と前記点Ｑを組み合わせてkQに対応する倍数の複合表現を形成するステップと、

ｄ）kQの前記複合表現から前記点kQに対応する値を算出するステップとを含むことを特徴とする方法。

【請求項２】　前記各成分$k_i$が、前記スカラーｋよりも短いことを特徴とする、請求項１に記載の方法。

【請求項３】　前記成分$k_i$がまず選択され、その後組み合わされて前記スカラーｋが形成されることを特徴とする、請求項１に記載の方法。

【請求項４】　前記表現が

【数１】

$$k_i = \sum_{i=0}^{i=} k_i \lambda^i \bmod n$$

の形式であり、ｎが楕円曲線上の点の数であることを特徴とする、請求項１に記載の方法。

【請求項５】　前記表現が$k_0 + k_1$の形式であることを特徴とする、請求項４に記載の方法。

【請求項６】　前記スカラーｋが所定の値および前記成分ｋを有することを特徴とする、請求項１に記載の方法。

【請求項７】　前記倍数kQの前記値が、同時多重加算を使用して算出されることを特徴とする、請求項３に記載の方法。

【請求項８】　前記同時多重加算において使用される一群の項$G_1$が事前に算

出されることを特徴とする、請求項７に記載の方法。

【請求項９】　体Ｆの短基底ベクトル$(u_0, u_1)$を得て、ベクトルｖを$(k, 0)$として指定し、ｖを正規直交基底から$(u_0, u_1)$基底に変換し、ベクトルｖを表す分数$f_0, f_1$を得て、前記分数をｋに適用しベクトルｚを得て、ベクトルｖの効率的な等価物v'を算出し、ベクトルv'の成分をkQの複合表現で使用することによって、前記成分$k_i$が得られることを特徴とする、請求項６に記載の方法。

【請求項10】　Ｑを曲線上の点として、整数ｋを有し、秘密鍵および公開鍵kQを提供する鍵対を楕円曲線暗号システムにおいて生成する方法であって、

ａ）楕円曲線上のすべての点$Q(x, y)$について$\psi(Q) = \lambda \cdot Q$であり、$\lambda$が整数である自己準同形写像$\psi$が存在するように、有限体Ｆの上の楕円曲線を選択するステップと、

ｂ）前記スカラーｋの表現を成分$k_i$と前記整数$\lambda$の組合わせとして確立するステップと、

ｃ）前記表現と前記点Ｑを組み合わせてkQに対応する倍数の複合表現を形成するステップと、

ｄ）kQの前記複合表現から前記点kQに対応する値を算出するステップとを含むことを特徴とする方法。

【請求項11】　請求項２から９のいずれか一項に記載の方法を含むことを特徴とする、請求項10に記載の方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、楕円曲線を利用して暗号システムで計算を実行する方法に関する。

【０００２】

【従来の技術】

公開鍵データ通信システムを使用して一対の通信者の間で情報を転送することができる。交換される情報の少なくとも一部は、送信側により所定の演算によって暗号化され、受信側は、送信側の演算に対して相補的な演算を実行してこの情報を復号することができる。

【０００３】

各通信者は、秘密鍵と、秘密鍵に数学的に関係づけられた公開鍵とを有する。この関係は、公開鍵の知識から秘密鍵を特定することができないような関係である。各鍵は、転送すべきデータを暗号化するために、あるいはデータが真正であることを検証できるように署名を添付するために、データの転送時に使用される。

【０００４】

暗号化の場合、一方の通信者は、受信側の公開鍵を使用してメッセージを暗号化し、受信側に送信する。次いで、受信側は秘密鍵を使用してメッセージを復号する。

【０００５】

一方の当事者の公開鍵を他方の当事者の秘密鍵と組み合わせることによって共通鍵を生成することもできる。このような場合には通常、各当事者の長期鍵が破られるのを回避するため、通信セッションごとに新しい秘密鍵およびこれに対応する公開鍵が生成され、これらの鍵は通常、セッション鍵または短命鍵と呼ばれる。

【０００６】

したがって、メッセージの交換および公開鍵の生成では、暗号化システムZ*pにおいて整数mod p（pは素数）の有限体を利用する際に指数演算を伴い、あるい

はシステムが楕円曲線を利用する際に同様な点乗算演算を伴う相当量の計算が行われる。楕円曲線システムでは、秘密整数ｋを生成し、シード点Ｑで点乗算を実行して短命公開鍵kQを形成することによって、短命鍵対が得られる。同様に、共通短命セッション鍵を生成する場合は、公開鍵$k_3Q$、すなわち、曲線上の点に他方の通信者の秘密整数$k_b$を乗じる必要があり、したがって、この場合も点乗算が必要になる。

【０００７】

メッセージに署名する場合にも、送信側が自分の秘密鍵をメッセージに適用することを除いて、同様な手順が使用される。この場合、任意の受信側が、送信側の公開鍵を使用してメッセージを復元し検証することができる。

【０００８】

このような方式を実施するための様々なプロトコルが存在しており、そのうちのいくつかは広く使用されている。しかし、それぞれの場合に、送信側は転送すべき情報に署名するために計算を実行する必要があり、受信側は、署名された情報を検証するために計算を実行する必要がある。

【０００９】

典型的な実行形式において、署名成分は以下の形式を有する。

$s = ae + k \pmod n$

上式で、楕円曲線暗号化システムにおいて、

Ｐは、基本曲線上の、システムの定義済みパラメータである点であり、

ｋは、短期秘密鍵またはセッション鍵として選択されたランダム整数であり、

$R = kP$は、これに対応する短期公開鍵であり、

ａは、送信側の長期秘密鍵であり、

$Q = aP$は、これに対応する送信側の公開鍵であり、

ｅは、メッセージｍおよび短期公開鍵Ｒの、SHA-1ハッシュ関数などの安全ハッシュであり、および

ｎは曲線の次数である。

【００１０】

送信側は、ｍ、ｓ、およびＲを含むメッセージを受信側に送信し、署名は、Ｒ

に対応すべき値$R^1$ = (sP - eQ)を算出することによって検証される。算出された値が対応する場合、署名は検証されたことになる。

【発明が解決しようとする課題】

【００１１】

検証を実行する場合、点乗算を計算し、それぞれ計算が複雑なsPおよびeQを得る必要がある。この場合、受信側が適切な計算力を有する場合には、特に問題はないが、セキュアトークン・アプリケーションや「スマート・カード」アプリケーションのように受信側の計算力が限られている場合、このような計算によって検証プロセスに遅延が生じる。

【００１２】

したがって、キー生成プロトコルおよび署名プロトコルは、大量の計算を必要とする場合がある。暗号化が普及するにつれて、より高速であり、スマート・カードや無線装置に見られるように限られた計算力を活用する暗号化システムを実現する要求が高まっている。

【００１３】

楕円曲線暗号化（ECC）は、この計算問題を解決する。ECCでは、鍵および証明書のサイズを縮小することができ、それによって、必要なメモリが低減し、コストが著しく節約される。ECCは、コストを著しく削減することができるだけでなく、次世代アプリケーションにおけるスマート・カードの普及を促進する。また、ECCアルゴリズムによって鍵サイズを縮小することができるが、より大きな鍵を用いる他のアルゴリズムと同じレベルのセキュリティが維持される。

【００１４】

しかし、暗号化装置の低い生産コストを維持しながら情報転送速度を高めるために、依然として、鍵に対する計算をより高速に行う必要がある。

【００１５】

楕円曲線上の点の倍数を算出することは、楕円曲線暗号化で最も頻繁に実行される計算の１つである。このような計算の速度を高める１つの方法は、事前計算された点の倍数のテーブルを使用することである。この技法は、点が事前にわかっているときにさらに有用である。しかし、未知の点の倍数が必要になる場合が

ある（たとえば、ECDSA検証）。したがって、点乗算を容易にするシステムおよび方法が必要である。

【００１６】

【課題を解決するための手段】

一般に、本発明は、スカラーｋを、成分$k_j$と、基本曲線で自己準同形写像から導かれる整数λとの組合わせとして表す。

【００１７】

この方法は、有限体の上に写像される複素乗算を有する楕円曲線（EC）が与えられた場合、複素乗算写像と点Qにλを乗じることが等しくなる二次方程式の解λが存在する、という考えに基づく方法である。λを整数とみなしてEC乗算を実行するのと比べて、複素乗算写像を介してλQを算出する方がコストが低いことが多い。実際には、他のスカラー（λ以外）による点乗算が必要になる。乗算写像を使用して点の他の倍数を算出できることも示す。

【００１８】

本発明によれば、楕円曲線点Q(x, y)にスカラーｋを乗じる速度を高める方法であって、

楕円曲線上のすべての点Q(x, y)について$\psi(Q) = \lambda \cdot Q$である自己準同形写像$\psi$が存在するように、有限体Ｆの上の楕円曲線を選択するステップと、

スカラーｋのより小さな表現$k_j$と写像$\psi$との組合わせを使用して楕円曲線点Ｑのスカラー倍数を算出するステップとを含む方法が提供される。

【００１９】

本発明の好ましい実施形態のこれらおよび他の特徴は、添付の図面が参照される以下の詳細な説明でより明らかになろう。

【００２０】

【発明の実施の形態】

以下の説明では、同じ符号は各図面中の同じ構造物を指す。図１を参照すると、データ通信システム10は、通信チャネル16によって接続され、送信側12および受信側14として指定された一対の通信者を含む。各通信者12、14は、デジタル情報を処理し、この情報を後述のようにチャネル16を通して送信する準備をするこ

とのできる暗号化プロセッサ18、20をそれぞれ含む。プロセッサ18、20は、プロセッサに組み込まれた集積回路で実現するか、あるいは汎用プロセッサと共に所定のプロトコルを実施するようにデータキャリア上に符号化された命令として実現することができる。図をわかりやすくするために、通信者12は、比較的限られた計算力を持つ専用プロセッサ18を有するスマート・カードの形であると仮定する。好ましくは、プロセッサ20は、チャネル16によってカードと通信する中央サーバであり、チャネル16は無線通信チャネルである。

【００２１】

暗号化プロセッサ18は、ECCの楕円曲線暗号化システムを実現し、また暗号化プロセッサ18の機能の１つは、整数であるｋと、基本楕円曲線上の点であるＱとを暗号化方式における鍵対k、kQとして使用できるように、ｋ・Ｑの形の点乗算を実行することである。上記で指摘したように、楕円曲線点とスカラー値の乗算などの暗号化計算はコストがかかる。

【００２２】

楕円曲線点Q(x, y)のスカラー乗算の速度を高める方法は、図２に示されており、全体的に符号50によって示されている。本発明のアルゴリズムは、プロセッサ12がたとえば、特定の種類の楕円曲線に関してメッセージに署名しメッセージを検証する速度を高める。この方法は、$F_q$（qは素数べき乗）として例示された有限体上の楕円曲線Ｅに関する以下の一般的な数式が与えられ、

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \qquad (1)$$

自己準同形写像$\psi$が存在し、楕円曲線上のすべての点Q(x, y)点について$\psi(Q) = \lambda \cdot Q$である場合、点Ｑと整数ｋとの乗算は、ｋのより小さな表現$k_j$と写像$\psi$との組合わせを利用することによって速度を高めることができるという考えに基づく方法である。写像$\psi$は、kQに関する後の計算で使用できる群要素およびそれらの組合わせを事前に算出することも可能にする。

【００２３】

次に図２を参照すると、楕円曲線上の点乗算の速度を高める一般的な実施形態のフローチャートが符号50で示されている。まずシステム・パラメータが選択される。最初のステップとして、ある特性を有する基本楕円曲線Ｅが選択される。

本発明の第１の実施形態では、一般化された楕円曲線（１）を以下の形式で表すことができる。

E: $y^2 = x^3 + b \bmod p$（Ｐは素数）　　（２）

【００２４】

まず、$\gamma \in F_p$（Fpはすべての整数mod pから成るサイズｐの体である）であり、かつ$\gamma^3 \equiv 1 \bmod p$（１の立方根）である数$\gamma$が存在するように係数ｐを求めることができる。たとえば、p = 7である場合、$2^3 \bmod 7 = 1$であるので、$\gamma$ = 2である。このような$\gamma$はすべてのｐに対して存在するわけではなく、したがって、ｐの値を選択する際にはこのことを考慮しなければならない。通常、適切な暗号強度を得るには、選択されるｐの長さが少なくとも160ビットであるべきである。

【００２５】

曲線Ｅが選択された後、写像関数$\psi$が求められる。写像関数$\psi$：(x, y)→($\gamma$x, y)は、曲線上の点のある組を曲線上の点の別の組に写像するに過ぎない。楕円曲線Ｅ上の当該のすべての点Q(x, y)点について$\psi(Q) = \lambda \cdot Q$になるような整数$\lambda$が存在する。この整数$\lambda$は、$\lambda^3 \equiv 1 \bmod n$であることに留意することによって求めることができる。この場合、ｎは、$F_p$の上にある楕円曲線Ｅ上の点の数、すなわち、$E(F_p)$上の点の数である。$\lambda^3 \equiv 1$の$\lambda$に複数の解が存在する場合があるが、写像関数$\psi$を満たす解は１つだけである。$\gamma^3 \bmod p = 1$であるので、Qと$\psi(Q)$の両方がＥに関する数式を満足する。したがって、長い計算を実行して$\lambda$による乗算の結果を求める代わりに、$\lambda$による乗算を非常に効率的に実行できるように写像関数の結果を使用して非常に効率的にこれを行うことができる。

【００２６】

シード点Ｑが選択され、52に示すように、製造時に、システム・パラメータＥ、ｐ、Ｑ、$\lambda$、$\psi(Q)$、および$\gamma$が、暗号化プロセッサ18によって使用できるようにカード12に記憶される。暗号化、鍵一致、署名などの暗号化手順を実施するには、短命秘密鍵ｋとして使用される整数ｋを選択し、対応する公開鍵kQを生成する必要がある。

【００２７】

ｋの値は次式ように表すことができる。

$k = (k_0 + k_1 \lambda) \bmod n$ （３）

【００２８】

上式で、ｎはE($F_p$)上の点の数であり、$k_0$および$k_1$は整数である。この場合、点k・Qは次式のようになる。

$k \cdot Q = (k_0 Q + k_1 \lambda Q) \bmod n$ （４）

【００２９】

いくつかの暗号演算の場合、ｋの値をランダムに選択することができ、このような場合、ｋを選択するのではなく、符号ビットを含まない長さ$[\log_2(n)]/2$の、$k_0$および$k_1$の値をランダムに選択し（すなわち、$k_1$の長さとして、長さｋの少なくとも２分の１の長さが選択される）、次いで数式（３）を使用してｋの値を算出することが可能である。図２に54で示すように$k_0$、$k_1$の値を選択した後、Menezesらによって「Handbook of Applied Cryptography」（HAC）に記載された「Simultaneous Multiple Exponentiation」（アルゴリズム14.88）に類似しており、56に示されているアルゴリズムを使用して数式（４）の右辺を高速に計算することができる。都合上、このアルゴリズムを以下に再現する。加群において、べき乗が加算に類似しており、したがって、このアルゴリズム内の乗算を加算で置き換えると以下の手順が得られる。

【００３０】

<u>アルゴリズム１　同時多重加算</u>

入力：群要素$g_0$, $g_1$, … $g_{l-1}$および負でないｔビット整数$e_0$, $e_1$, … $e_{l-1}$。

出力：$g_0e_0 + g_1e_1 + \cdots + g_{l-1}e_{l-1}$。

ステップ１。事前計算。ｉが０から$(2^l - 1)$に対して、

【数２】

$$G_i \leftarrow \Sigma_{j=0}^{l-1} g_j i_j$$

上式で、$i = (i_{l-1} \ldots i_0)_2$である。

ステップ２。A←0

ステップ３。ｉが１からｔに達するまで、以下のことを実行する。

A←A+A、A←A+$G_l$

ステップ４。（Ａ）を返す。この場合、$A = g_0e_0 + g_1e_1 + \cdots + g_{l-1}e_{l-1}$

【００３１】

このアルゴリズムを数式（４）に適用すると、２つの群要素$g_0$、$g_1$、すなわちＱおよびλＱがあり、したがって、２つの整数$e_0$、$e_1$、すなわち$k_0$、$k_1$があることがわかる。このアルゴリズムでは、いくつかの値を事前に算出することができ、最初に$G_i$が算出される。$l = 2$によって$G_i$を事前に算出した結果をテーブル１（表１）に示す。

【表１】

| i | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $G_i$ | 0 | $g_0$ | $g_1$ | $g_0+g_1$ |

【００３２】

点加算：（$Q + \psi(Q)$）を実行して点を作成した後、算出済みの要素をテーブル１に記入してテーブル２（表２）を作成することが可能である。図２のステップ58に示すように、これらの要素を事前に算出しメモリに記憶することができる。

【表２】

| i | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $G_i$ | 0 | Q | $\psi$（Q） | Q＋$\psi$（Q） |

$G_{l_i}$を求め、したがって、60に示すように$l_1$から$l_t$を求めないかぎり、アルゴリズムのステップを実行することはできない。$k_i$の２進表現を使用して概念行列または組合わせテーブルを作成することができる。たとえば、$k_0 = 30$であり$k_1 = 10$である場合、$k_0$から$k_1$の２進表現内の最大ビット数が５であるのでｔは値５を有する。$k_0$から$k_1$の２進表現で作成された概念行列をテーブル３に示す。$l_i$は、第１の行が最下位ビットを含み、第２の行が次の下位ビットを含み、以下同様であるｉ番目の列に表された数によって決定される。したがって、テーブル３（表３）から、$l_1 = l_2 = (11) = 3$、$l_3 = (01) = 1$、$l_4 = 3$、および$l_5 = 0$であることがわかる。

【表３】

| i | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $k_0$ | 1 | 1 | 1 | 1 | 0 |
| $k_1$ | 0 | 1 | 0 | 1 | 0 |
| $l_i$ | 1 | 3 | 1 | 3 | 0 |

【００３３】

このアルゴリズムを完了するのに必要なすべての成分を得ることができ、62に示すようにステップ３の反復が行われる。

【００３４】

最初はA←0であり、ｉは１に設定される。$l_i = l_1$であり、これはテーブル３から１に等しい。したがって、$G_{l1}$は$G_1$であり、これはテーブル２からＱである。したがって、l = 1の場合の反復によるＡの値は0 + Q = Qである。

【００３５】

i = 2である次の反復の場合、Ａの初期値はＱであり、したがって、A← Q+Q = 2Qであり、テーブル３から$l_i = l_2 = 3$である。したがって、$G_{l2}$はテーブル３から$G_3$に等しく、すなわち、Q + $\psi$(Q)である。

【００３６】

したがって、A + $G_{li}$は2Q + Q +$\psi$Q = 3Q +$\psi$Qと算出される。

【００３７】

反復は、５回目の反復まで、すなわち、koq = k, λQの値、すなわちkQが算出されるまで、テーブル４（表４）に記載されたｉの値ごとに継続される。

【表４】

| i | A |
|---|---|
| 1 | Q |
| 2 | ３Q＋$\psi$（Q） |
| 3 | ７Q＋２$\psi$（Q） |
| 4 | １５Q＋５（Q） |
| 5 | ３０Q＋１０$\psi$（Q） |

【００３８】

各反復では、点倍加（A+A）および点加算（A+$G_{li}$）が必要である。ただし、場

合によっては、$G_{|i}$の値は０であり、この場合、計算量が削減される。

【００３９】

したがって、この方法では、$\max\{\log_2(ki)\}$に等しい数の点倍加と、ほぼ同数の点加算が必要であることがわかる。点加算の数は、ウィンドウ技法（アルゴリズム14.85 HAC）および指数再符号化技法を使用して削減することができる。iおよび$G_i$の値を事前に算出することができるので、事前に算出された適切な要素$G_i$をテーブル２から検索することによって点加算を容易に実行することができる。kPが算出された後、チャネル16を介した暗号化伝送または署名伝送において通信者12の短命公開鍵としてこのkPを使用することができる。

【００４０】

簡単に言えば、暗号化やDH署名などの暗号化演算の場合、整数ｋが必要であり、対応する公開鍵kQが算出される。それぞれ、長さｎの２分の１の長さを有する、値$k_0$および$k_1$がランダムに選択され、適切なアルゴリズムを使用して項$k_0Q = k_1\lambda Q$が生成される。ｋをこのように選択すると、この方法は、ｋ自体をランダムに生成するのと同程度に安全であると思われる。もちろん、効率を向上するためにより少ないビット数の$k_i$を選択することが可能である。

【００４１】

上記の技法において、$k = k_0 + k_1\lambda$を書き込み、同時に組み合わせる方法によって、同時多重加算アルゴリズムの速度が高められる。$k = k_0 + k_1\lambda$を書き込む技法は、スカラー乗算技法、すなわち、ワインディング、組合わせなどと共に使用することもでき有利である。

【００４２】

いくつかの写像$\psi$では、３つ以上の下位要素ｋ（sub k）を使用することも可能である。いくつかの写像$\psi$では、$k = k_0 + k_1\lambda + k_2\lambda^2$を書き込み、同時多重加算アルゴリズムを適用することによってｋの値を算出することができる。

【００４３】

本発明の第２の実施形態では、一般化された楕円曲線式（１）の異なる形式、すなわち、

$$y^2 = (x^3 - ax) \bmod p \qquad (5)$$

が使用される。この場合も、ｐは少なくとも160ビットを有する素数である。この種の曲線の場合、γに必要な特性が異なる。この場合、$\gamma^2 = -1 \bmod p$になるような値を求める必要がある。γの特性を変更するには、異なる写像関数ψ'を使用する必要がある。この実施形態では、写像は形式ψ' ：(x, y) → (-x, γy)をとる。(x, y)が曲線上にある場合はψ' (x, y)も曲線上にある。この場合、$\lambda^4 \equiv 1 \bmod n$（ｎはこの場合も、$E(F_p)$上の点である）、したがってλを算出することができる。写像ψ' (Q) = λ・Qは前述のように実行され、この場合も、この曲線に対してλによる乗算を非常に効率的に行うことができる。この実施形態のｋに関する数式は、第１の実施形態と同じであり、次式によって表される。

$k = (k_0 + k_1 \lambda) \bmod n$ （６）

この数式は、前の実施形態と同じであり、２つの群要素のみを有する。したがって、アルゴリズム１の群要素Ｑおよび$Q+\psi'(Q)$を使用して点ｋ・Ｑを算出することができる。この計算では、$\max\{\log_2(k_i)\}$に等しい数の点倍加と、同様な数の点加算が必要である。前述のように、ウィンドウ技法および指数再符号化技法を使用して点加算の数を削減することができる。

【００４４】

この方法は、効率的に算出できる自己準同形写像ψが存在するかぎり、他の楕円曲線にも適用することができる。

【００４５】

上記の実施形態は、ｋをランダムに選択することができ、したがって、$k_0$および$k_1$を選択することができるものと仮定し、ｋを求めている。ｋを選択することが可能な暗号化プロトコルの場合、まず、$k = (k_0 + k_1 \lambda) \bmod n$になるようにｋの所与の値から所望の「短い」形式の$k_0$、$k_1$を求める必要がある。場合によっては、３つ以上のｋを使用することができ有利である。

【００４６】

上述の実施形態でわかるように、事前に点がわかっているときは、テーブルを作成して乗算の速度を高めることができる。しかし、未知の点の倍数が必要になる場合があり（たとえば、これはECDSA検証で起こる可能性がある）、その場合

、与えられたｋの値をとり、次いで$k_i$の適切な表現を決定する必要がある。

【００４７】

したがって、第３の実施形態では、システム・パラメータおよび値ｋが与えられ、点Ｑ、必要な倍数ｋ、および複素乗算倍数λが既知である。所定のｋの値から「短い」$k_i$を求める必要がある。これを行う方法を以下に説明し、図３のフローチャートに示す。（ｋを必要としない）事前計算として、$a_i$および$b_i$がｎよりも小さな数になるような２つの関係を算出する。

$a_0 + b_0\lambda = 0 \bmod n$

$a_1 + b_1\lambda = 0 \bmod n$

$a_i$および$b_i$はできるだけ小さいことが好ましいが、本発明の方法は、$a_i$および$b_i$が最小限でないときでも有利である。$a_i$および$b_i$が共に小さな対$a_i$および$b_i$は、小さなユークリッド長を有するベクトル$u_i$とみなすことができる。通常、後述の方法では、最初のｋのサイズの２分の１の表現を有する$k_0$および$k_1$が生成される。

【００４８】

本実施形態では、事前に算出された短いベクトル表現を使用して以下の形式の数式を得ることによって、kQを効率的に算出することができる。

$k_0Q + \lambda k_1Q$

【００４９】

これは、事前に算出されたベクトルを使用して、ｋの知識を必要としない分数$f_0$および$f_1$を導くことによって行われる。ベクトルｚは、分数$f_0$および$f_1$とｋを組み合わせることによって生成される。ベクトルｚを使用して、$v' = (v_0', v_1')$である第２のベクトルv'が算出され、kQの値が、

【数３】

$$v_o'Q + \lambda v_1'Q \qquad (8)$$

として算出される。この解を得る方法を以下に詳しく説明する。

【００５０】

小さな$a_i$および$b_i$を作成する場合、$L_3$、すなわち、短い基底ベクトルが直接得

られる格子基底削減アルゴリズム（HAC、118ページ）を利用することが可能である。しかし、この好ましい実施形態では、簡単な拡張ユークリッド・アルゴリズムが対（n, λ）に対して使用される。（n, λ）に対する拡張ユークリッド・アルゴリズムにより、iに応じて$r_i$の表現（たとえば、ビット長）が小さくなり、$c_i$および$d_i$の表現が大きくなる線形組合わせ$c_i n + d_i \lambda = r_i$が作成される。

【００５１】

拡張ユークリッド・アルゴリズムを使用した結果として得られた$|(d_i, r_i)|$の２つの最小値が保存される。これらのベクトルのサイズが、平方ユークリッド基準$|(d_i, r_i)| = d_i^2 + r_i^2$を用いて測定される。これらの最小関係の項は、

$$\hat{d}_0, \hat{r}_0$$

および

$$\hat{d}_1, \hat{r}_1$$

として示され、通常、アルゴリズムの中央で得られる。最小関係が保持されない場合でも、この方法は、部分最適関係により、依然として点倍数の計算において有利である。

【００５２】

$a_i$および$b_i$の値は、すべて事前に算出することのできる

$$a_0 = -\hat{r}_0, \; b_0 = \hat{d}_0$$

および

$$a_1 = -\hat{r}_1, \; b_1 = \hat{d}_0$$

を定義することによって作成することができる。

【００５３】

次のタスクは、倍数kの小さな表現を求めることである。

【００５４】

$a_0$、$b_0$および$a_1$, $b_1$の計算が与えられた場合、$u_0 = (a_0, b_0)$であり$u_1 = (a_1,$

$b_1$)であるベクトル$u_0$, $u_1$を指定することが可能である。これらのベクトルは$a_i + b_i \lambda = 0 \pmod n$を満たす。群要素Qにベクトルv = ($v_0$, $v_1$)を乗じることは($v_0 + v_1 \lambda$)Qとして定義される。$a_i + b_i \lambda = 0 \pmod n$であるので、任意の群要素Rについて$u_0R = u_1R = 0$が成立する。したがって、任意の整数$z_0$および$z_1$のの場合、任意の群要素Rについて$v' R = (v - z_0u_0 - z_1u_1)R$が成立する。

【００５５】

整数$z_0$および$z_1$としては、ベクトル$v' = v - z_0u_0 - z_1u_1$ができるだけ小さな成分を有するような整数が選択される。この場合も、この方法は、v'の成分が小さい場合に利点を有し、該成分が必ずしも最小限でない場合でも利点を有する。

【００５６】

適切な$z_0$および$z_1$は、vの基底を正規直交基底{$u_0$, $u_1$}に変換することによって算出される。基底間の変換では行列乗算が行われる。ベクトルv = (v0, v1)を{u0, u1}基底から標準基底{(1, 0), (0, 1)}に変換する場合、次式が成立する。

【数４】

$$v_{\{(1,0),(0,1)\}} = v_{(u_0,u_1)}M = (v_0, v_1)\begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \end{bmatrix}$$

他方の方向、すなわち、正規直交基底{(1, 0), (0, 1)}から($u_0$, $u_1$)基底に変換する場合、乗算は単にMの逆数によって行われる。

【数５】

$$v_{(u_0,u_1)} = v_{\{(1,0),(0,1)\}}inverse(M) = v_{\{(1,0),(0,1)\}}\frac{1}{a_0b_1 - a_1b_0}\begin{bmatrix} b_1 & -b_0 \\ -a_1 & a_0 \end{bmatrix}$$

【００５７】

ベクトルv = (k, 0)がゼロ成分を有するので、逆数（M）の下の行は必要とされず、したがって、{$u_0$, $u_1$}に変換する場合に必要なのは次式の分数だけである。

【数６】

$$f_0 = \frac{b_1}{a_0 b_1 - a_1 b_0}$$

および

【数７】

$$f_1 = \frac{b_0}{a_0 b_1 - a_1 b_0}$$

【００５８】

分数$f_0$および$f_1$の演算を乗算のみで行うことができるように、これらの分数を十分な精度に事前に算出しておくことができる。これらの分数をもたらす計算がｋに依存せず、したがって、楕円曲線がシステム・パラメータとして選択される際にこれらの分数を１度算出することができ、各ｋごとに再計算する必要はない。同様に、ベクトルv、$u_0$、および$u_1$を事前に算出し記憶することができる。

【００５９】

ｋの値が選択されるか、あるいは求められた後、まず、ｚが$(z_0, z_1)$ = (round$(kf_0)$, round$(kf_1)$)として定義されるz = $(z_0, z_1)$を計算することによってkQの値を算出することができる。ｚの近傍の他のベクトルも有用であり、したがって、丸めを床関数または天井関数、あるいは何らかの他の近似で置き換えることができる。

【００６０】

適切なｚが求められた後、v' = $(v_0', v_1')$ = v - $z_0u_0$ - $z_1u_1$によってv(k, 0)の効率的な等価物が算出される。「効率的な等価物」の句は、v' P = vPおよびv'が小さな係数を有するようなベクトルv'を意味する。この場合、値kQは$v_0'$ Q + $v_1'$ λQとして算出される。この値は、上述のように同時点加算を使用して算出することができ、上記で説明し、かつH.A.C.14.7の627ページに記載されているように非隣接形式（NAF）再符号化を使用することによってより高い効率を得ることができる。したがって、ｋが所定の値である場合でも、$k_0$および$k_1$の値を算出し写像関数と共に使用してkQの値、したがって、鍵対k、kQを得ることができる。

【００６１】

ｋを３つの部分$k = k_0 + k_1\lambda + k_2\lambda^2$に分離する場合、$L^3$行削減によって、次式のような小さなベクトルを得ることができる。

【数８】

$$\begin{bmatrix} 1 & 0 & -\lambda^2 \\ 0 & 1 & -\lambda \\ 0 & 0 & -n \end{bmatrix} \text{to} \quad \begin{bmatrix} u_2 \\ u_1 \\ u_0 \end{bmatrix}$$

【００６２】

二次元の場合と同様に小ベクトル等価物（三次元行）を得ることができる。

【００６３】

これらの方法を使用してk・Qの値を求めると、暗号化プロセッサ12によって必要とされる処理能力が大幅に低減される。また、このような反復計算が行われる速度が高められ、それによって、情報を転送するための時間が短縮される。

【００６４】

当然のことながら、スカラー倍数ｋが短縮成分$k = k_0 + k_1\lambda + k_2\lambda^2 + \ldots k_{m-1}\lambda^{m-1}$で表された後、同時多重加算アルゴリズムの代わりに、あるいはこのアルゴリズムと共に、効率的な楕円曲線スカラー乗算を行うための他の方法を使用することができる。このような方法には、ウィンドウ技法（固定およびスライド）、組合わせ技法、ビット再符号化技法、およびこれらの技法の組合わせが含まれる。

【００６５】

特に有益なある技法では、乗算の一成分、たとえば$k_0$に関して作成されたテーブルを他の成分$k_1$などに再使用することができる。これは、必要に応じて写像γを適用することにより、算出されたテーブル要素を変換することによって行われる。

【００６６】

他の例として、ｋを$k = k_0 + k_1\lambda + k_2\lambda^2$として再設定することができ、ｋがｍビットを有し、$k_i$が約m/3ビットを有する実施形態について以下に説明する

【００６７】

成分$k_i$は、求められた後、２進表現から、より少ない非ゼロ・ビットを有する符号付き２進表現に再符号化することができる。この再符号化は、表現$k_i$内のあらゆる１ビットまたは－１ビットが符号付き２進文字列内で他の非ゼロに隣接しないような非隣接形式（NAF）をとることができる。

【００６８】

各$k_i$が再符号化された後、$k_i\lambda^i P$の計算を助けるためにテーブルを作成することができる。

【００６９】

NAFウィンドウ・テーブルは、$\lambda^i P$のある短ビット長倍数を事前に算出するためのテーブルである。このウィンドウの幅によってテーブルのサイズが決定される。$k_i$が、隣接する非ゼロを有さないように再符号化されているので、奇数ウィンドウ幅が適切である。３ビット幅NAFウィンドウは以下のものを含む。

【数９】

| 1 | 101 | 10-1 |
|---|---|---|

【００７０】

再符号化$k_i$は、このようなウィンドウを連結し、必要に応じてゼロを充填することによって作成される（H.A.C.、616ページ）。

【００７１】

あらゆる非ゼロ・ビットではなく、発生するあらゆるウィンドウごとにのみEC点を加算または減算するだけでよいので、このテーブルを使用することによって必要な加算数を削減することができる。

【００７２】

したがって、まず、この技法は$k_0P$の計算に適用される。

【００７３】

$k_0P$の計算のために作成されたテーブルは、演算子$\gamma$を使用してテーブル要素

に$\psi$写像が写像される場合に$k_i \lambda P$計算に適用することができる。同様に、$k_0P$用に作成されたテーブルを使用するが、テーブル要素に$\gamma^2$を写像することによって、$k_2 \lambda^2 P$の速度を高めることができる。

【００７４】

スライド・ウィンドウ技法を該成分に適用する際に、実行する必要があるのは１組の倍加だけである。

【００７５】

好ましい実施形態のこの例を示すために、以下の例を使用する。

【００７６】

$k = [101101011101]_2 + [111010101101]_2 \lambda$である場合、

以下の再符号化を行う。

$k = [10\text{-}100\text{-}10\text{-}100\text{-}101] + [1000\text{-}10\text{-}10\text{-}10\text{-}101]\lambda$

$= k'_0 + k'_1 \lambda$

【００７７】

1・P、[10-1]・P、[101]・Pを含む、Pに関する３ビット・ウィンドウ・テーブルが事前に算出される。この場合、２回のEC加算および２回のEC倍加が必要である。

【００７８】

この後、テーブルの要素を加算／減算することによって、kPを以下のように算出することができる。

$kP = [10\text{-}100\text{-}10\text{-}100\text{-}101]P + [1000\text{-}10\text{-}10\text{-}10\text{-}101] \cdot \lambda P$

これは、累算器Ａを使用して以下のように行うことができる。

A←0　；初期設定

$A \mathrel{+}= \psi(1 \cdot P)$　；$k_0'$の上位ビットを消費する。

$A \leftarrow 2A$　；Ａを倍加する。

$A \leftarrow 2A$

$A \leftarrow [10\text{-}1]P$　；$k_0'$の上位３ビットを消費する。

$A \leftarrow 2^4 A$

$A \mathrel{-}= [101]\psi P$　；$k_1'$の３ビット・ウィンドウを消費する。

A ← 2A　；Ａを倍加する。

A - = [101]P　；$k_1$’の３ビットを消費する。

A ← $2^4$A

A - = [101]$\psi$P　；$k_1$’の３ビットを消費する。

A ← $2^2$A

A - = [10-1]P　；$k_0$’の最後のビットを消費する。

A + =$\psi$P　；kPを生成する。

【００７９】

簡単に言えば、前述の技法は以下のとおりである。楕円曲線Ｅおよび自己準同形写像$\psi$が与えられた場合、すべての点Q∈Eについて$\lambda Q = \psi(Q)$になるような対応する整数$\lambda$が存在する。整数ｍを選択し、同等な数ｍの「短基底ベクトル」$b_1, b_2, \ldots, b_m$を算出する。このような基底ベクトルはそれぞれ整数に対応し、そのような整数はそれぞれ、点の数$n = \#E(F_p^m)$（すなわち、点の数）で除することができる。次に、整数ｋ（0 <; k <; n）が与えられた場合、$k_i$として「短い」ベクトルが選択される$k = \Sigma k_i \cdot \lambda^i$と書くことができる。これは、$b_1, b_2, \cdots, b_m$によって生成された格子内の（ｋを表す）あるベクトルと近傍のベクトルとの差を求めることによって行われる。

【００８０】

以下の実施形態では、複合体の上に画定された楕円曲線に前述の技法（自己準同形写像および基底変換および「Shamir’ｓ trick」）を適用することについて明白に説明する。特に、ｐが奇素数である曲線$E(F_p^m)$への適用について説明する。以下の実施形態ではこのような曲線に対する技法を例示する。

【００８１】

この技法について、写像$\psi$がフロベニウス写像$\psi(x, y) = (x^p, y^p)$であり、A, B∈$F_p$であるE’$_{A,B}(F_p^m)$が使用される場合において説明する。

【００８２】

この場合、フロベニウス写像が$\psi^2 - t\psi + p = 0$を満たし、$t = p + 1 - \#E(F_p^m)$であることがわかっている。

【００８３】

したがって、$\lambda^2 - t\lambda + p = 0 \bmod n$であり、また$\lambda^{2+i} - t\lambda^{1+i} + p\lambda^i = 0 \bmod n$である。

【００８４】

以下のベクトルが、ベクトル空間Qn空間のｍ個の「短」基底ベクトルで構成されている。

【数１０】

$$
\begin{array}{ll}
 & (\lambda^{m-1}\ldots \qquad \lambda^2, \lambda^1, \lambda^0) \\
b_1 & (0,\ 0,\ 0,\ldots \quad 0,\ 1,\ -t,\ p) \\
b_2 & (\qquad\qquad\quad 1,\ -t,\ p,\ 0) \\
 & \\
 & (1,\ -t,\ p,0,0,\ldots \qquad \ldots,0) \\
 & (-t,\ p,\ 0,0,\ldots \qquad \ldots,0,\ 1) \\
b_m & (p,\ 0,\ 0,0,\ldots \quad 0,\ 1,\ -t)
\end{array}
$$

したがって、このような曲線に対するk・Qを計算する場合、ベクトル$b_1$, $b_2$, … $b_m$および前述の技法を使用することができる。

【００８５】

上記の実施形態では、k, λQを$\psi$(kQ)から得ることができ、写像が加算よりも効率的であることが理解されよう。

【００８６】

本発明を特定の実施形態を参照して説明したが、当業者には、添付の請求の範囲に記載された本発明の趣旨および範囲から逸脱しない本発明の様々な修正形態が明らかになろう。

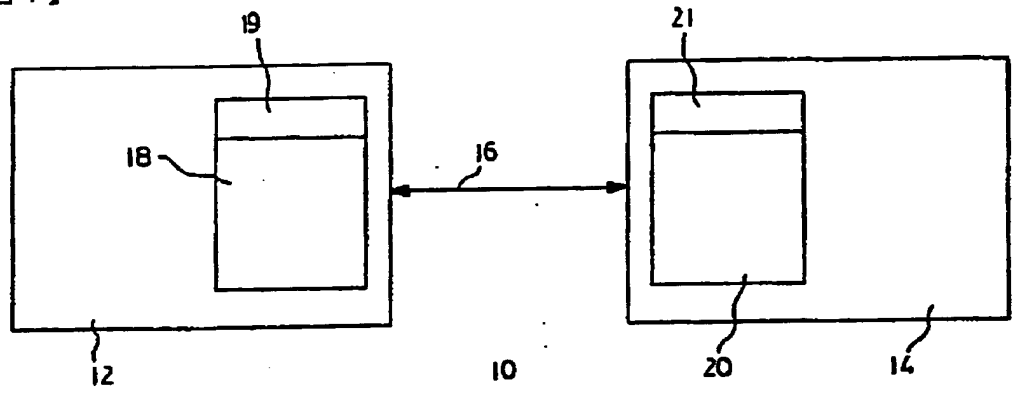
【図面の簡単な説明】

【図１】　通信システムの概略図である。

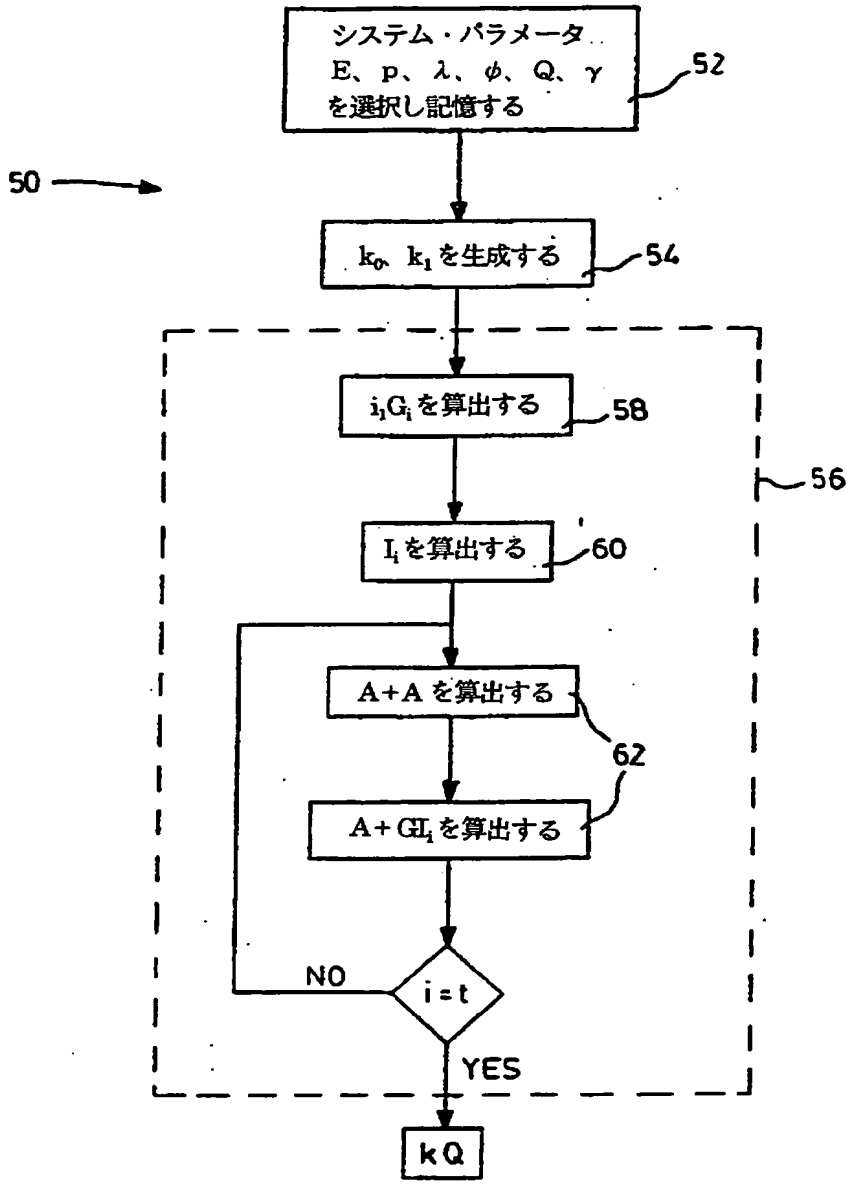
【図２】　本発明の第１の実施形態を実施する各ステップを示すフローチャートである。

【図３】　図２の方法を実施するために必要なパラメータを与えるステップを示すフローチャートである。

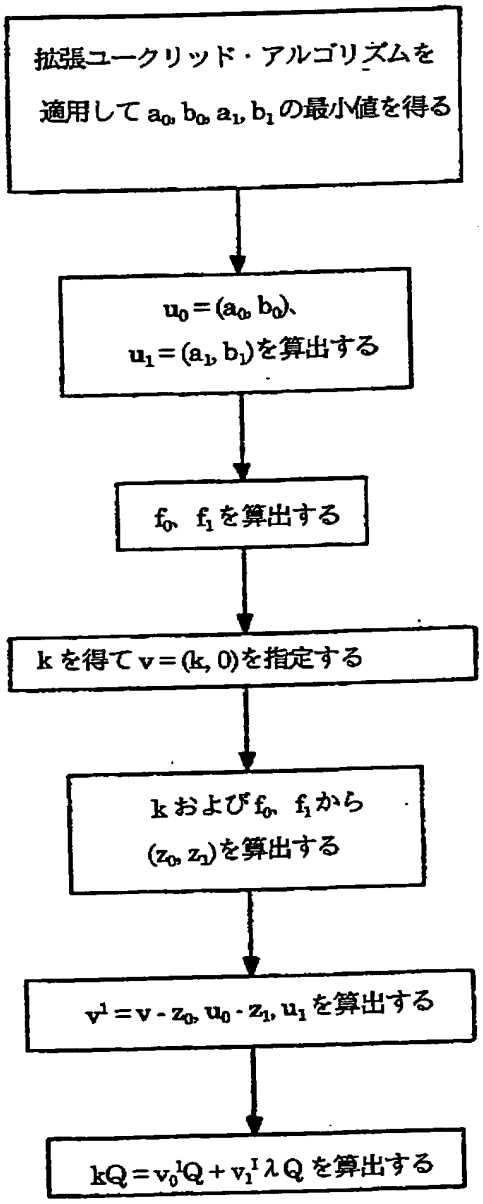
【図１】



【図２】


システム・パラメータ
E、p、λ、φ、Q、γ
を選択し記憶する
52
50
$k_0$、$k_1$を生成する
54
$i_1G_i$を算出する
58
56
$I_i$を算出する
60
A+A を算出する
62
A+$GI_i$を算出する
NO
i=t
YES
kQ

【図３】


拡張ユークリッド・アルゴリズムを適用して $a_0, b_0, a_1, b_1$ の最小値を得る
$u_0=(a_0, b_0)$、$u_1=(a_1, b_1)$を算出する
$f_0$、$f_1$を算出する
kを得てv=(k, 0)を指定する
kおよび$f_0$、$f_1$から$(z_0, z_1)$を算出する
$v^1=v-z_0, u_0-z_1, u_1$を算出する
$kQ=v_0^1Q+v_1^1\lambda Q$を算出する

【手続補正書】

【提出日】平成１３年８月１０日（２００１．８．１０）

【手続補正１】

【補正対象書類名】明細書

【補正対象項目名】発明の詳細な説明

【補正方法】変更

【補正の内容】

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、楕円曲線を利用して暗号システムで計算を実行する方法に関する。

【０００２】

【従来の技術】

公開鍵データ通信システムを使用して一対の通信者の間で情報を転送することができる。交換される情報の少なくとも一部は、送信側により所定の演算によって暗号化され、受信側は、送信側の演算に対して相補的な演算を実行してこの情報を復号することができる。

【０００３】

各通信者は、秘密鍵と、秘密鍵に数学的に関係づけられた公開鍵とを有する。この関係は、公開鍵の知識から秘密鍵を特定することができないような関係である。各鍵は、転送すべきデータを暗号化するために、あるいはデータが真正であることを検証できるように署名を添付するために、データの転送時に使用される。

【０００４】

暗号化の場合、一方の通信者は、受信側の公開鍵を使用してメッセージを暗号化し、受信側に送信する。次いで、受信側は秘密鍵を使用してメッセージを復号する。

【０００５】

一方の当事者の公開鍵を他方の当事者の秘密鍵と組み合わせることによって共

通鍵を生成することもできる。このような場合には通常、各当事者の長期鍵が破られるのを回避するため、通信セッションごとに新しい秘密鍵およびこれに対応する公開鍵が生成され、これらの鍵は通常、セッション鍵または短命鍵と呼ばれる。

【０００６】

したがって、メッセージの交換および公開鍵の生成では、暗号化システム$Z^*_p$において整数mod p（pは素数）の有限体を利用する際に指数演算を伴い、あるいはシステムが楕円曲線を利用する際に同様な点乗算演算を伴う相当量の計算が行われる。楕円曲線システムでは、秘密整数ｋを生成し、シード点Ｑで点乗算を実行して短命公開鍵kQを形成することによって、短命鍵対が得られる。同様に、共通短命セッション鍵を生成する場合は、公開鍵$k_3Q$、すなわち、曲線上の点に他方の通信者の秘密整数$k_b$を乗じる必要があり、したがって、この場合も点乗算が必要になる。

【０００７】

メッセージに署名する場合にも、送信側が自分の秘密鍵をメッセージに適用することを除いて、同様な手順が使用される。この場合、任意の受信側が、送信側の公開鍵を使用してメッセージを復元し検証することができる。

【０００８】

このような方式を実施するための様々なプロトコルが存在しており、そのうちのいくつかは広く使用されている。しかし、それぞれの場合に、送信側は転送すべき情報に署名するために計算を実行する必要があり、受信側は、署名された情報を検証するために計算を実行する必要がある。

【０００９】

典型的な実行形式において、署名成分は以下の形式を有する。

$s = ae + k \pmod n$

上式で、楕円曲線暗号化システムにおいて、

Ｐは、基本曲線上の、システムの定義済みパラメータである点であり、

ｋは、短期秘密鍵またはセッション鍵として選択されたランダム整数であり、

$R = kP$は、これに対応する短期公開鍵であり、

ａは、送信側の長期秘密鍵であり、
Ｑ＝ａＰは、これに対応する送信側の公開鍵であり、
ｅは、メッセージｍおよび短期公開鍵Ｒの、SHA-1ハッシュ関数などの安全ハッシュであり、および
ｎは曲線の次数である。

【００１０】

送信側は、ｍ、ｓ、およびＲを含むメッセージを受信側に送信し、署名は、Ｒに対応すべき値$R^1 = (sP - eQ)$を算出することによって検証される。算出された値が対応する場合、署名は検証されたことになる。

【発明が解決しようとする課題】

【００１１】

検証を実行する場合、点乗算を計算し、それぞれ計算が複雑なsPおよびeQを得る必要がある。この場合、受信側が適切な計算力を有する場合には、特に問題はないが、セキュアトークン・アプリケーションや「スマート・カード」アプリケーションのように受信側の計算力が限られている場合、このような計算によって検証プロセスに遅延が生じる。

【００１２】

したがって、キー生成プロトコルおよび署名プロトコルは、大量の計算を必要とする場合がある。暗号化が普及するにつれて、より高速であり、スマート・カードや無線装置に見られるように限られた計算力を活用する暗号化システムを実現する要求が高まっている。

【００１３】

楕円曲線暗号化（ECC）は、この計算問題を解決する。ECCでは、鍵および証明書のサイズを縮小することができ、それによって、必要なメモリが低減し、コストが著しく節約される。ECCは、コストを著しく削減することができるだけでなく、次世代アプリケーションにおけるスマート・カードの普及を促進する。また、ECCアルゴリズムによって鍵サイズを縮小することができるが、より大きな鍵を用いる他のアルゴリズムと同じレベルのセキュリティが維持される。

【００１４】

しかし、暗号化装置の低い生産コストを維持しながら情報転送速度を高めるために、依然として、鍵に対する計算をより高速に行う必要がある。

【００１５】

楕円曲線上の点の倍数を算出することは、楕円曲線暗号化で最も頻繁に実行される計算の１つである。このような計算の速度を高める１つの方法は、事前計算された点の倍数のテーブルを使用することである。この技法は、点が事前にわかっているときにさらに有用である。しかし、未知の点の倍数が必要になる場合がある（たとえば、ECDSA検証）。したがって、点乗算を容易にするシステムおよび方法が必要である。

【００１６】

【課題を解決するための手段】

一般に、本発明は、スカラーｋを、成分$k_i$と、基本曲線で自己準同形写像から導かれる整数λとの組合わせとして表す。

【００１７】

この方法は、有限体の上に写像される複素乗算を有する楕円曲線（EC）が与えられた場合、複素乗算写像と点Ｑにλを乗じることが等しくなる二次方程式の解λが存在する、という考えに基づく方法である。λを整数とみなしてEC乗算を実行するのと比べて、複素乗算写像を介してλQを算出する方がコストが低いことが多い。実際には、他のスカラー（λ以外）による点乗算が必要になる。乗算写像を使用して点の他の倍数を算出できることも示す。

【００１８】

本発明によれば、楕円曲線点Q(x, y)にスカラーｋを乗じる速度を高める方法であって、

楕円曲線上のすべての点Q(x, y)について$\psi(Q) = \lambda \cdot Q$である自己準同形写像$\psi$が存在するように、有限体Ｆの上の楕円曲線を選択するステップと、

スカラーｋのより小さな表現$k_i$と写像$\psi$との組合わせを使用して楕円曲線点Ｑのスカラー倍数を算出するステップとを含む方法が提供される。

【００１９】

本発明の好ましい実施形態のこれらおよび他の特徴は、添付の図面が参照され

る以下の詳細な説明でより明らかになろう。

【００２０】

【発明の実施の形態】

以下の説明では、同じ符号は各図面中の同じ構造物を指す。図１を参照すると、データ通信システム10は、通信チャネル16によって接続され、送信側12および受信側14として指定された一対の通信者を含む。各通信者12、14は、デジタル情報を処理し、この情報を後述のようにチャネル16を通して送信する準備をすることのできる暗号化プロセッサ18、20をそれぞれ含む。プロセッサ18、20は、プロセッサに組み込まれた集積回路で実現するか、あるいは汎用プロセッサと共に所定のプロトコルを実施するようにデータキャリア上に符号化された命令として実現することができる。図をわかりやすくするために、通信者12は、比較的限られた計算力を持つ専用プロセッサ18を有するスマート・カードの形であると仮定する。好ましくは、プロセッサ20は、チャネル16によってカードと通信する中央サーバであり、チャネル16は無線通信チャネルである。

【００２１】

暗号化プロセッサ18は、ECCの楕円曲線暗号化システムを実現し、また暗号化プロセッサ18の機能の１つは、整数であるｋと、基本楕円曲線上の点であるＱとを暗号化方式における鍵対k、kQとして使用できるように、ｋ・Qの形の点乗算を実行することである。上記で指摘したように、楕円曲線点とスカラー値の乗算などの暗号化計算はコストがかかる。

【００２２】

楕円曲線点Q(x, y)のスカラー乗算の速度を高める方法は、図２に示されており、全体的に符号50によって示されている。本発明のアルゴリズムは、プロセッサ12がたとえば、特定の種類の楕円曲線に関してメッセージに署名しメッセージを検証する速度を高める。この方法は、$F_q$（qは素数べき乗）として例示された有限体上の楕円曲線Ｅに関する以下の一般的な数式が与えられ、

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \qquad (1)$$

自己準同形写像ψが存在し、楕円曲線上のすべての点Q(x, y)点についてψ(Q) = λ・Qである場合、点Ｑと整数ｋとの乗算は、ｋのより小さな表現$k_i$と写像ψと

の組合わせを利用することによって速度を高めることができるという考えに基づく方法である。写像$\psi$は、kQに関する後の計算で使用できる群要素およびそれらの組合わせを事前に算出することも可能にする。

【００２３】

次に図２を参照すると、楕円曲線上の点乗算の速度を高める一般的な実施形態のフローチャートが符号50で示されている。まずシステム・パラメータが選択される。最初のステップとして、ある特性を有する基本楕円曲線Ｅが選択される。本発明の第１の実施形態では、一般化された楕円曲線（１）を以下の形式で表すことができる。

$E: y^2 = x^3 + b \bmod p$（Ｐは素数）　　（２）

【００２４】

まず、$\gamma \in F_p$（Fpはすべての整数mod pから成るサイズｐの体である）であり、かつ$\gamma^3 \equiv 1 \bmod p$（１の立方根）である数$\gamma$が存在するように係数ｐを求めることができる。たとえば、$p = 7$である場合、$2^3 \bmod 7 = 1$であるので、$\gamma = 2$である。このような$\gamma$はすべてのｐに対して存在するわけではなく、したがって、ｐの値を選択する際にはこのことを考慮しなければならない。通常、適切な暗号強度を得るには、選択されるｐの長さが少なくとも160ビットであるべきである。

【００２５】

曲線Ｅが選択された後、写像関数$\psi$が求められる。写像関数$\psi$：$(x, y) \rightarrow (\gamma x, y)$は、曲線上の点のある組を曲線上の点の別の組に写像するに過ぎない。楕円曲線Ｅ上の当該のすべての点Q(x, y)点について$\psi(Q) = \lambda \cdot Q$になるような整数$\lambda$が存在する。この整数$\lambda$は、$\lambda^3 \equiv 1 \bmod n$であることに留意することによって求めることができる。この場合、ｎは、$F_p$の上にある楕円曲線Ｅ上の点の数、すなわち、$E(F_p)$上の点の数である。$\lambda^3 \equiv 1$の$\lambda$に複数の解が存在する場合があるが、写像関数$\psi$を満たす解は１つだけである。$\gamma^3 \bmod p = 1$であるので、Qと$\psi(Q)$の両方がＥに関する数式を満足する。したがって、長い計算を実行して$\lambda$による乗算の結果を求める代わりに、$\lambda$による乗算を非常に効率的に実行できるように写像関数の結果を使用して非常に効率的にこれを行うことができる。

【００２６】

シード点Ｑが選択され、52に示すように、製造時に、システム・パラメータＥ、ｐ、Ｑ、λ、ψ(Q)、およびγが、暗号化プロセッサ18によって使用できるようにカード12に記憶される。暗号化、鍵一致、署名などの暗号化手順を実施するには、短命秘密鍵ｋとして使用される整数ｋを選択し、対応する公開鍵kQを生成する必要がある。

【００２７】

ｋの値は次式ように表すことができる。

$k = (k_0 + k_1 \lambda) \bmod n$　　（３）

【００２８】

上式で、ｎはE($F_p$)上の点の数であり、$k_0$および$k_1$は整数である。この場合、点k・Qは次式のようになる。

$k \cdot Q = (k_0 Q + k_1 \lambda Q) \bmod n$　　（４）

【００２９】

いくつかの暗号演算の場合、ｋの値をランダムに選択することができ、このような場合、ｋを選択するのではなく、符号ビットを含まない長さ$[\log_2(n)]/2$の、$k_0$および$k_1$の値をランダムに選択し（すなわち、$k_1$の長さとして、長さｋの少なくとも２分の１の長さが選択される）、次いで数式（３）を使用してｋの値を算出することが可能である。図２に54で示すように$k_0$、$k_1$の値を選択した後、Menezesらによって「Handbook of Applied Cryptography」（HAC）に記載された「Simultaneous Multiple Exponentiation」（アルゴリズム14.88）に類似しており、56に示されているアルゴリズムを使用して数式（４）の右辺を高速に計算することができる。都合上、このアルゴリズムを以下に再現する。加群において、べき乗が加算に類似しており、したがって、このアルゴリズム内の乗算を加算で置き換えると以下の手順が得られる。

【００３０】

<u>アルゴリズム１　同時多重加算</u>

入力：群要素$g_0, g_1, \cdots g_{l-1}$および負でないｔビット整数$e_0, e_1, \cdots e_{l-1}$。

出力：$g_0e_0 + g_1e_1 + \cdots + g_{l-1}e_{l-1}$。

ステップ１。事前計算。ｉが０から$(2^l - 1)$に対して、

【数２】

$$G_i \leftarrow \Sigma_{j=0}^{l-1} g_j{}^{i_j}$$

上式で、$i = (i_{l-1} \ldots i_0)_2$である。

ステップ２。A←0

ステップ３。ｉが１からｔに達するまで、以下のことを実行する。

A←A+A、A←A+$G_l$

ステップ４。（Ａ）を返す。この場合、$A = g_0e_0 + g_1e_1 + \cdots + g_{l-1}e_{l-1}$

【００３１】

このアルゴリズムを数式（４）に適用すると、２つの群要素$g_0$、$g_1$、すなわちＱおよびλＱがあり、したがって、２つの整数$e_0$、$e_1$、すなわち$k_0$、$k_1$があることがわかる。このアルゴリズムでは、いくつかの値を事前に算出することができ、最初に$G_i$が算出される。l = 2によって$G_i$を事前に算出した結果をテーブル１（表１）に示す。

【表１】

| j | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $G_i$ | 0 | $g_0$ | $g_1$ | $g_0+g_1$ |

【００３２】

点加算：（Ｑ＋ψ(Ｑ)）を実行して点を作成した後、算出済みの要素をテーブル１に記入してテーブル２（表２）を作成することが可能である。図２のステップ58に示すように、これらの要素を事前に算出しメモリに記憶することができる。

【表２】

| i | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $G_i$ | 0 | Ｑ | ψ（Ｑ） | Ｑ＋ψ（Ｑ） |

$G_{|i}$を求め、したがって、60に示すように$l_1$から$l_t$を求めないかぎり、アルゴリ

ズムのステップを実行することはできない。$k_i$の２進表現を使用して概念行列または組合わせテーブルを作成することができる。たとえば、$k_0 = 30$であり$k_1 = 10$である場合、$k_0$から$k_1$の２進表現内の最大ビット数が５であるのでｔは値５を有する。$k_0$から$k_1$の２進表現で作成された概念行列をテーブル３に示す。$l_i$は、第１の行が最下位ビットを含み、第２の行が次の下位ビットを含み、以下同様であるｉ番目の列に表された数によって決定される。したがって、テーブル３（表３）から、$l_1 = l_2 = (11) = 3$、$l_3 = (01) = 1$、$l_4 = 3$、および$l_5 = 0$であることがわかる。

【表３】

| i | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $k_0$ | 1 | 1 | 1 | 1 | 0 |
| $k_1$ | 0 | 1 | 0 | 1 | 0 |
| $I_i$ | 1 | 3 | 1 | 3 | 0 |

【００３３】

このアルゴリズムを完了するのに必要なすべての成分を得ることができ、62に示すようにステップ３の反復が行われる。

【００３４】

最初はA←0であり、ｉは１に設定される。$l_i = l_1$であり、これはテーブル３から１に等しい。したがって、$G_{l1}$は$G_1$であり、これはテーブル２からＱである。したがって、$l = 1$の場合の反復によるＡの値は$0 + Q = Q$である。

【００３５】

$i = 2$である次の反復の場合、Ａの初期値はＱであり、したがって、A← $Q+Q = 2Q$であり、テーブル３から$l_i = l_2 = 3$である。したがって、$G_{l2}$はテーブル３から$G_3$に等しく、すなわち、$Q + \psi(Q)$である。

【００３６】

したがって、$A + G_{li}$は$2Q + Q + \psi Q = 3Q + \psi Q$と算出される。

【００３７】

反復は、５回目の反復まで、すなわち、$k_{oq} = k, \lambda Q$の値、すなわちkQが算出されるまで、テーブル４（表４）に記載されたｉの値ごとに継続される。

【表４】

| i | A |
|---|---|
| 1 | Q |
| 2 | ３Ｑ＋ψ（Ｑ） |
| 3 | ７Ｑ＋２ψ（Ｑ） |
| 4 | １５Ｑ＋５（Ｑ） |
| 5 | ３０Ｑ＋１０ψ（Ｑ） |

【００３８】

各反復では、点倍加（A+A）および点加算（$A+G_{|i}$）が必要である。ただし、場合によっては、$G_{|i}$の値は０であり、この場合、計算量が削減される。

【００３９】

したがって、この方法では、$\max\{\log_2(k_i)\}$に等しい数の点倍加と、ほぼ同数の点加算が必要であることがわかる。点加算の数は、ウィンドウ技法（アルゴリズム14.85 HAC）および指数再符号化技法を使用して削減することができる。iおよび$G_i$の値を事前に算出することができるので、事前に算出された適切な要素$G_i$をテーブル２から検索することによって点加算を容易に実行することができる。kPが算出された後、チャネル16を介した暗号化伝送または署名伝送において通信者12の短命公開鍵としてこのkPを使用することができる。

【００４０】

簡単に言えば、暗号化やDH署名などの暗号化演算の場合、整数ｋが必要であり、対応する公開鍵kQが算出される。それぞれ、長さｎの２分の１の長さを有する、値$k_0$および$k_1$がランダムに選択され、適切なアルゴリズムを使用して項$k_0Q = k_1\lambda Q$が生成される。ｋをこのように選択すると、この方法は、ｋ自体をランダムに生成するのと同程度に安全であると思われる。もちろん、効率を向上するためにより少ないビット数の$k_i$を選択することが可能である。

【００４１】

上記の技法において、$k = k_0 + k_1\lambda$を書き込み、同時に組み合わせる方法によって、同時多重加算アルゴリズムの速度が高められる。$k = k_0 + k_1\lambda$を書き込む技法は、スカラー乗算技法、すなわち、ワインディング、組合わせなどと共に使用することもでき有利である。

【００４２】

いくつかの写像$\psi$では、３つ以上の下位要素ｋ（sub k）を使用することも可能である。いくつかの写像$\psi$では、$k = k_0 + k_1\lambda + k_2\lambda^2$を書き込み、同時多重加算アルゴリズムを適用することによってｋの値を算出することができる。

【００４３】

本発明の第２の実施形態では、一般化された楕円曲線式（１）の異なる形式、すなわち、

$y^2 = (x^3 - ax) \bmod p$　　（５）

が使用される。この場合も、ｐは少なくとも160ビットを有する素数である。この種の曲線の場合、$\gamma$に必要な特性が異なる。この場合、$\gamma^2 = -1 \bmod p$になるような値を求める必要がある。$\gamma$の特性を変更するには、異なる写像関数$\psi'$を使用する必要がある。この実施形態では、写像は形式$\psi'$：$(x, y) \rightarrow (-x, \gamma y)$をとる。$(x, y)$が曲線上にある場合は$\psi'(x, y)$も曲線上にある。この場合、$\lambda^4 \equiv 1 \bmod n$（ｎはこの場合も、$E(F_p)$上の点である）、したがってλを算出することができる。写像$\psi'(Q) = \lambda \cdot Q$は前述のように実行され、この場合も、この曲線に対してλによる乗算を非常に効率的に行うことができる。この実施形態のｋに関する数式は、第１の実施形態と同じであり、次式によって表される。

$k = (k_0 + k_1\lambda) \bmod n$　　（６）

この数式は、前の実施形態と同じであり、２つの群要素のみを有する。したがって、アルゴリズム１の群要素Ｑおよび$Q+\psi'(Q)$を使用して点$k \cdot Q$を算出することができる。この計算では、$\max\{\log_2(k_i)\}$に等しい数の点倍加と、同様な数の点加算が必要である。前述のように、ウィンドウ技法および指数再符号化技法を使用して点加算の数を削減することができる。

【００４４】

この方法は、効率的に算出できる自己準同形写像$\psi$が存在するかぎり、他の楕円曲線にも適用することができる。

【００４５】

上記の実施形態は、ｋをランダムに選択することができ、したがって、$k_0$およ

び$k_1$を選択することができるものと仮定し、ｋを求めている。ｋを選択することが可能な暗号化プロトコルの場合、まず、$k = (k_0 + k_1\lambda) \bmod n$になるようにｋの所与の値から所望の「短い」形式の$k_0$、$k_1$を求める必要がある。場合によっては、３つ以上のｋを使用することができ有利である。

【００４６】

上述の実施形態でわかるように、事前に点がわかっているときは、テーブルを作成して乗算の速度を高めることができる。しかし、未知の点の倍数が必要になる場合があり（たとえば、これはECDSA検証で起こる可能性がある）、その場合、与えられたｋの値をとり、次いで$k_i$の適切な表現を決定する必要がある。

【００４７】

したがって、第３の実施形態では、システム・パラメータおよび値ｋが与えられ、点Ｑ、必要な倍数ｋ、および複素乗算倍数λが既知である。所定のｋの値から「短い」$k_i$を求める必要がある。これを行う方法を以下に説明し、図３のフローチャートに示す。（ｋを必要としない）事前計算として、$a_i$および$b_i$がｎよりも小さな数になるような２つの関係を算出する。

$a_0 + b_0\lambda = 0 \bmod n$

$a_1 + b_1\lambda = 0 \bmod n$

$a_i$および$b_i$はできるだけ小さいことが好ましいが、本発明の方法は、$a_i$および$b_i$が最小限でないときでも有利である。$a_i$および$b_i$が共に小さな対$a_i$および$b_i$は、小さなユークリッド長を有するベクトル$u_i$とみなすことができる。通常、後述の方法では、最初のｋのサイズの２分の１の表現を有する$k_0$および$k_1$が生成される。

【００４８】

本実施形態では、事前に算出された短いベクトル表現を使用して以下の形式の数式を得ることによって、kQを効率的に算出することができる。

$k_0Q + \lambda k_1Q$

【００４９】

これは、事前に算出されたベクトルを使用して、ｋの知識を必要としない分数$f_0$および$f_1$を導くことによって行われる。ベクトルｚは、分数$f_0$および$f_1$とｋを

組み合わせることによって生成される。ベクトルｚを使用して、$v' = (v_0', v_1')$である第２のベクトルv'が算出され、kQの値が、

【数３】

$$vo'Q + \lambda v_1'Q \qquad (8)$$

として算出される。この解を得る方法を以下に詳しく説明する。

【００５０】

小さな$a_i$および$b_i$を作成する場合、$L_3$、すなわち、短い基底ベクトルが直接得られる格子基底削減アルゴリズム（HAC、118ページ）を利用することが可能である。しかし、この好ましい実施形態では、簡単な拡張ユークリッド・アルゴリズムが対（n, λ）に対して使用される。（n, λ）に対する拡張ユークリッド・アルゴリズムにより、ｉに応じて$r_i$の表現（たとえば、ビット長）が小さくなり、$c_i$および$d_i$の表現が大きくなる線形組合わせ$c_i n + d_i \lambda = r_i$が作成される。

【００５１】

拡張ユークリッド・アルゴリズムを使用した結果として得られた$|(d_i, r_i)|$の２つの最小値が保存される。これらのベクトルのサイズが、平方ユークリッド基準$|(d_i, r_i)| = d_i^2 + r_i^2$を用いて測定される。これらの最小関係の項は、

$$\hat{d}_0, \hat{r}_0$$

および

$$\hat{d}_1, \hat{r}_1$$

として示され、通常、アルゴリズムの中央で得られる。最小関係が保持されない場合でも、この方法は、部分最適関係により、依然として点倍数の計算において有利である。

【００５２】

$a_i$および$b_i$の値は、すべて事前に算出することのできる

$$a_0 = -\hat{r}_0, b_0 = \hat{d}_0$$

および

$$a_1 = -\hat{r}_1, \quad b_1 = \hat{d}_0$$

を定義することによって作成することができる。

【００５３】

次のタスクは、倍数ｋの小さな表現を求めることである。

【００５４】

$a_0$、$b_0$および$a_1$，$b_1$の計算が与えられた場合、$u_0 = (a_0, b_0)$であり$u_1 = (a_1, b_1)$であるベクトル$u_0$，$u_1$を指定することが可能である。これらのベクトルは$a_i + b_i \lambda = 0 \pmod n$を満たす。群要素Ｑにベクトル$v = (v_0, v_1)$を乗じることは$(v_0 + v_1 \lambda)Q$として定義される。$a_i + b_i \lambda = 0 \pmod n$であるので、任意の群要素Ｒについて$u_0R = u_1R = 0$が成立する。したがって、任意の整数$z_0$および$z_1$のの場合、任意の群要素Ｒについて$v' R = (v - z_0u_0 - z_1u_1)R$が成立する。

【００５５】

整数$z_0$および$z_1$としては、ベクトル$v' = v - z_0u_0 - z_1u_1$ができるだけ小さな成分を有するような整数が選択される。この場合も、この方法は、v’の成分が小さい場合に利点を有し、該成分が必ずしも最小限でない場合でも利点を有する。

【００５６】

適切な$z_0$および$z_1$は、vの基底を正規直交基底$\{u_0, u_1\}$に変換することによって算出される。基底間の変換では行列乗算が行われる。ベクトル$v = (v0, v1)$を$\{u0, u1\}$基底から標準基底$\{(1, 0), (0, 1)\}$に変換する場合、次式が成立する。

【数４】

$$v_{\{(1,0),(0,1)\}} = v_{(u_0,u_1)}M = (v_0, v_1)\begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \end{bmatrix}$$

他方の方向、すなわち、正規直交基底{(1, 0), (0, 1)}から($u_0$, $u_1$)基底に変換する場合、乗算は単にMの逆数によって行われる。

【数５】

$$v_{(u_0,u_1)} = v_{\{(1,0),(0,1)\}}inverse(M) = v_{\{(1,0),(0,1)\}}\frac{1}{a_0b_1 - a_1b_0}\begin{bmatrix} b_1 & -b_0 \\ -a_1 & a_0 \end{bmatrix}$$

【００５７】

ベクトルv = (k, 0)がゼロ成分を有するので、逆数（M）の下の行は必要とされず、したがって、{$u_0$, $u_1$}に変換する場合に必要なのは次式の分数だけである。

【数６】

$$f_0 = \frac{b_1}{a_0b_1 - a_1b_0}$$

および

【数７】

$$f_1 = \frac{b_0}{a_0b_1 - a_1b_0}$$

【００５８】

分数$f_0$および$f_1$の演算を乗算のみで行うことができるように、これらの分数を十分な精度に事前に算出しておくことができる。これらの分数をもたらす計算がkに依存せず、したがって、楕円曲線がシステム・パラメータとして選択される

際にこれらの分数を１度算出することができ、各ｋごとに再計算する必要はない。同様に、ベクトルv、$u_0$、および$u_1$を事前に算出し記憶することができる。

【００５９】

ｋの値が選択されるか、あるいは求められた後、まず、ｚが$(z_0, z_1)$ = (round$(kf_0)$, round$(kf_1)$)として定義されるz = $(z_0, z_1)$を計算することによってkQの値を算出することができる。ｚの近傍の他のベクトルも有用であり、したがって、丸めを床関数または天井関数、あるいは何らかの他の近似で置き換えることができる。

【００６０】

適切なｚが求められた後、$v' = (v_0', v_1') = v - z_0u_0 - z_1u_1$によってv(k, 0)の効率的な等価物が算出される。「効率的な等価物」の句は、v' P = vPおよびv'が小さな係数を有するようなベクトルv'を意味する。この場合、値kQは$v_0' Q + v_1' \lambda Q$として算出される。この値は、上述のように同時点加算を使用して算出することができ、上記で説明し、かつH. A. C. 14. 7の627ページに記載されているように非隣接形式（NAF）再符号化を使用することによってより高い効率を得ることができる。したがって、ｋが所定の値である場合でも、$k_0$および$k_1$の値を算出し写像関数と共に使用してkQの値、したがって、鍵対k、kQを得ることができる。

【００６１】

ｋを３つの部分$k = k_0 + k_1\lambda + k_2\lambda^2$に分離する場合、$L^3$行削減によって、次式のような小さなベクトルを得ることができる。

【数８】

$$\begin{bmatrix} 1 & 0 & -\lambda^2 \\ 0 & 1 & -\lambda \\ 0 & 0 & -n \end{bmatrix} \text{ to } \begin{bmatrix} u_2 \\ u_1 \\ u_0 \end{bmatrix}$$

【００６２】

二次元の場合と同様に小ベクトル等価物（三次元行）を得ることができる。

【００６３】

これらの方法を使用してk・Qの値を求めると、暗号化プロセッサ12によって必要とされる処理能力が大幅に低減される。また、このような反復計算が行われる速度が高められ、それによって、情報を転送するための時間が短縮される。

【００６４】

当然のことながら、スカラー倍数ｋが短縮成分$k = k_0 + k_1\lambda + k_2\lambda^2 + \ldots k_{m-1}\lambda^{m-1}$で表された後、同時多重加算アルゴリズムの代わりに、あるいはこのアルゴリズムと共に、効率的な楕円曲線スカラー乗算を行うための他の方法を使用することができる。このような方法には、ウィンドウ技法（固定およびスライド）、組合わせ技法、ビット再符号化技法、およびこれらの技法の組合わせが含まれる。

【００６５】

特に有益なある技法では、乗算の一成分、たとえば$k_0$に関して作成されたテーブルを他の成分$k_1$などに再使用することができる。これは、必要に応じて写像γを適用することにより、算出されたテーブル要素を変換することによって行われる。

【００６６】

他の例として、ｋを$k = k_0 + k_1\lambda + k_2\lambda^2$として再設定することができ、ｋがｍビットを有し、$k_i$が約m/3ビットを有する実施形態について以下に説明する。

【００６７】

成分$k_i$は、求められた後、２進表現から、より少ない非ゼロ・ビットを有する符号付き２進表現に再符号化することができる。この再符号化は、表現$k_i$内のあらゆる１ビットまたは－1ビットが符号付き２進文字列内で他の非ゼロに隣接しないような非隣接形式（NAF）をとることができる。

【００６８】

各$k_i$が再符号化された後、$k_i\lambda^i P$の計算を助けるためにテーブルを作成することができる。

【００６９】

NAFウィンドウ・テーブルは、$\lambda^i P$のある短ビット長倍数を事前に算出するためのテーブルである。このウィンドウの幅によってテーブルのサイズが決定される。$k_i$が、隣接する非ゼロを有さないように再符号化されているので、奇数ウィンドウ幅が適切である。３ビット幅NAFウィンドウは以下のものを含む。

【数９】

| 1 | 101 | 10-1 |
|---|---|---|

【００７０】

再符号化$k_i$は、このようなウィンドウを連結し、必要に応じてゼロを充填することによって作成される（H.A.C.、616ページ）。

【００７１】

あらゆる非ゼロ・ビットではなく、発生するあらゆるウィンドウごとにのみEC点を加算または減算するだけでよいので、このテーブルを使用することによって必要な加算数を削減することができる。

【００７２】

したがって、まず、この技法は$k_0P$の計算に適用される。

【００７３】

$k_0P$の計算のために作成されたテーブルは、演算子$\gamma$を使用してテーブル要素に$\psi$写像が写像される場合に$k_i \lambda P$計算に適用することができる。同様に、$k_0P$用に作成されたテーブルを使用するが、テーブル要素に$\gamma^2$を写像することによって、$k_2 \lambda^2 P$の速度を高めることができる。

【００７４】

スライド・ウィンドウ技法を該成分に適用する際に、実行する必要があるのは１組の倍加だけである。

【００７５】

好ましい実施形態のこの例を示すために、以下の例を使用する。

【００７６】

$k = [1011010111101]_2 + [1110101011101]_2\lambda$ である場合、

以下の再符号化を行う。

$k = [10\text{-}100\text{-}10\text{-}100\text{-}101] + [1000\text{-}10\text{-}10\text{-}10\text{-}101]\lambda$

$= k'_0 + k'_1\lambda$

【００７７】

１・P、[10-1]・P、[101]・Pを含む、Pに関する３ビット・ウィンドウ・テーブルが事前に算出される。この場合、２回のEC加算および２回のEC倍加が必要である。

【００７８】

この後、テーブルの要素を加算／減算することによって、kPを以下のように算出することができる。

$kP = [10\text{-}100\text{-}10\text{-}100\text{-}101]P + [1000\text{-}10\text{-}10\text{-}10\text{-}101]\cdot\lambda P$

これは、累算器Ａを使用して以下のように行うことができる。

A←0　；初期設定

$A \mathrel{+}= \psi(1\cdot P)$　；$k_0'$ の上位ビットを消費する。

$A \leftarrow 2A$　；Ａを倍加する。

$A \leftarrow 2A$

$A \leftarrow [10\text{-}1]P$　；$k_0'$ の上位３ビットを消費する。

$A \leftarrow 2^4A$

$A \mathrel{-}= [101]\psi P$　；$k_1'$ の３ビット・ウィンドウを消費する。

$A \leftarrow 2A$　；Ａを倍加する。

$A \mathrel{-}= [101]P$　；$k_1'$ の３ビットを消費する。

$A \leftarrow 2^4A$

$A \mathrel{-}= [101]\psi P$　；$k_1'$ の３ビットを消費する。

$A \leftarrow 2^2A$

$A \mathrel{-}= [10\text{-}1]P$　；$k_0'$ の最後のビットを消費する。

$A \mathrel{+}= \psi P$　；kPを生成する。

【００７９】

簡単に言えば、前述の技法は以下のとおりである。楕円曲線Ｅおよび自己準同

形写像$\psi$が与えられた場合、すべての点Q∈Eについて$\lambda Q = \psi(Q)$になるような対応する整数$\lambda$が存在する。整数mを選択し、同等な数mの「短基底ベクトル」$b_1, b_2, \ldots, b_m$を算出する。このような基底ベクトルはそれぞれ整数に対応し、そのような整数はそれぞれ、点の数$n = \#E(F_p^m)$（すなわち、点の数）で除することができる。次に、整数k（$0 < k < n$）が与えられた場合、$k_i$として「短い」ベクトルが選択される$k = \Sigma k_i \cdot \lambda^i$と書くことができる。これは、$b_1, b_2, \cdots, b_m$によって生成された格子内の（kを表す）あるベクトルと近傍のベクトルとの差を求めることによって行われる。

【００８０】

以下の実施形態では、複合体の上に画定された楕円曲線に前述の技法（自己準同形写像および基底変換および「Shamir's trick」）を適用することについて明白に説明する。特に、pが奇素数である曲線$E(F_p^m)$への適用について説明する。以下の実施形態ではこのような曲線に対する技法を例示する。

【００８１】

この技法について、写像$\psi$がフロベニウス写像$\psi(x, y) = (x^p, y^p)$であり、A, B∈$F_p$である$E'_{A,B}(F_p^m)$が使用される場合において説明する。

【００８２】

この場合、フロベニウス写像が$\psi^2 - t\psi + p = 0$を満たし、$t = p + 1 - \#E(F_p^m)$であることがわかっている。

【００８３】

したがって、$\lambda^2 - t\lambda + p = 0 \bmod n$であり、また$\lambda^{2+i} - t\lambda^{1+i} + p\lambda^i = 0 \bmod n$である。

【００８４】

以下のベクトルが、ベクトル空間Qn空間のm個の「短」基底ベクトルで構成されている。

【数１０】

$$
\begin{array}{lll}
 & (\lambda^{m-1}\ldots & \lambda^2, \lambda^1, \lambda^0) \\
b_1 & (0,\ 0,\ 0,\ldots & 0,\ 1,\ -t,\ p) \\
b_2 & ( & 1, -t,\ p,\ 0) \\
 & & \\
 & (1,\ -t,\ p, 0, 0,\ldots & \ldots, 0) \\
 & (-t,\ p,\ 0, 0,\ \ldots & \ldots, 0,\ 1) \\
b_m & (p,\ 0,\ 0, 0,\ldots & 0,\ 1, -t)
\end{array}
$$

したがって、このような曲線に対するk・Qを計算する場合、ベクトル$b_1$, $b_2$, … $b_m$および前述の技法を使用することができる。

【００８５】

上記の実施形態では、k, $\lambda$ Qを$\psi$(kQ)から得ることができ、写像が加算よりも効率的であることが理解されよう。

【国際調査報告】

# INTERNATIONAL SEARCH REPORT

Int. onal Application No
PCT/CA 99/01222

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F7/72

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category * | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | KOBLITZ N: "CM-CURVES WITH GOOD CRYPTOGRAPHIC PROPERTIES" PROCEEDINGS OF THE CONFERENCE ON THEORY AND APPLICATIONS OF CRYPTOGRAPHIC TECHNIQUES (CRYPTO),DE,BERLIN, SPRINGER, vol. -, 1991, pages 279-287, XP000269035 page 280 — | 1-8 |
| A | V. MÜLLER: "Fast Multiplication on Elliptic Curves over Small Fields of Characteristic Two" -, 'Online! 30 June 1997 (1997-06-30), pages 1-19, XP002121579 Retrieved from the Internet: <URL:ftp://ftp.informatik.tu-darmstadt.de/pub/TI/reports/vmueller.jc.ps.gz> 'retrieved on 1999-10! the whole document — | 1-8 |

-/---

[X] Further documents are listed in the continuation of box C. [ ] Patent family members are listed in annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 2 May 2000 | 15/05/2000 |
| Name and mailing address of the ISA<br>European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,<br>Fax: (+31-70) 340-3016 | Authorized officer<br>Verhoof, P |

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int. onal Application No
PCT/CA 99/01222

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

| Category * | Citation of document, with indication,where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | CHEON J H ET AL: "Two efficient algorithms for arithmetic of elliptic curves using Frobenius map" PUBLIC KEY CRYPTOGRAPHY. FIRST INTERNATIONAL WORKSHOP ON PRACTICE AND THEORY IN PUBLIC KEY CRYPTOGRAPHY, PKC'98. PROCEEDINGS, PUBLIC KEY CRYPTOGRAPHY FIRST INTERNATIONAL WORKSHOP ON PRACTICE AND THEORY IN PUBLIC KEY CRYPTOGRAPHY, PKC'98 PROCEEDINGS, , pages 195-202, XP000905121 1998, Berlin, Germany, Springer-Verlag, Germany ISBN: 3-540-64693-0 the whole document | 1-8 |
| A | SOLINAS J A: "An improved algorithm for arithmetic on a family of elliptic curves" ADVANCES IN CRYPTOLOGY - CRYPTO '97. 17TH ANNUAL INTERNATIONAL CRYPTOLOGY CONFERENCE. PROCEEDINGS, ADVANCES IN CRYPTOLOGY - CRYPTO'97. 17TH ANNUAL INTERNATIONAL CRYPTOLOGY CONFERENCE. PROCEEDINGS, SANTA BARBARA, CA, USA, 17-21 AUG. 1997, pages 357-371, XP002136758 1997, Berlin, Germany, Springer-Verlag, Germany ISBN: 3-540-63384-7 the whole document | 1-8 |

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

フロントページの続き

(81)指定国 EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW

(72)発明者 ヴァンストーン、スコット、エー.
カナダ国 エルＯピー １ビーＯ オンタリオ州、キャンプベルヴィル、ピー. オー. ボックス 490、パインビュー トレイル 10140

Ｆターム(参考) 5J104 AA18 AA25 JA25 NA02 NA16 NA35 NA40